NPS52-80-004

NAVAL POSTGRADUATE SCHOOL
Monterey, California

REPRESENTATION OF DISCRETE OPTIMIZATION PROBLEMS
BY DISCRETE DYNAMIC PROGRAMS

Douglas R. Smith

February 1980

ed for public release; distribution unlimited

red for:
Postgraduate School
Monterey, Ca. 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral J. J. Ekelund
Superintendent

Jack R. Borsting
Provost

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-80-004	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Representation of Discrete Optimization Problems by Discrete Dynamic Programs		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Douglas R. Smith		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N; RR000-01-10 N0001480WR00054
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, Ca. 93940		12. REPORT DATE February 1980
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Discrete optimization, Dynamic programming, Problem representations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper investigaes the conditions under which a discrete optimization problem can be formulated as a dynamic program. Following the terminology of (Karp and Held, 1967), a discrete optimization problem is formalized as a discrete decision process and the class of dynamic programs is formalized as a sequential decision process. Necessary and sufficient conditions for the representation in two different senses of a discrete decision problem by a sequential decision process are established. In the first sense (a

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

A

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

strong representation) the set of all optimal solutions to the discrete optimization problem is obtainable from the solution of the functional equations of dynamic programming. In the second sense (a weak representation) of nonempty subset of optimal solutions is obtainable from the solution of the functional equations of dynamic programming. It is shown that the well known principle of optimality corresponds to a strong representation. A more general version of the principle of optimality is given which corresponds to a weak representation of a discrete decision problem by a sequential decision process. We also show that the class of strongly representable discrete decision problems is equivalent to the class of sequential decision processes which have cost functions satisfying a strict monotonicity condition. Also a new derivation is given of the result that the class of weakly representable discrete decision problems is equivalent to the class of sequential decision processes which have a cost function satisfying a monotonicity condition.

Representation of Discrete Optimization Problems by Discrete Dynamic Programs

Douglas R. Smith

Naval Postgraduate School
Monterey, California

The work reported herein was supported by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

C

ABSTRACT

This paper investigates the conditions under which a discrete optimization problem can be formulated as a dynamic program. Following the terminology of (Karp and Held 1967), a discrete optimization problem is formalized as a discrete decision problem and the class of dynamic programs is formalized as a sequential decision process. Necessary and sufficient conditions for the representation in two different senses of a discrete decision problem by a sequential decision process are established. In the first sense (a strong representation) the set of all optimal solutions to the discrete optimization problem is obtainable from the solution of the functional equations of dynamic programming. In the second sense (a weak representation) a nonempty subset of optimal solutions is obtainable from the solution of the functional equations of dynamic programming. It is shown that the well known principle of optimality corresponds to a strong representation. A more general version of the principle of optimality is given which corresponds to a weak representation of a discrete decision problem by a sequential decision process. We also show that the class of strongly representable discrete decision problems is equivalent to the class of sequential decision prcesses which have cost functions satisfying a strict monotonicity condition. Also a new derivation is given of the result that the class of weakly representable discrete decision problems is equivalent to the class of sequential decision processes which have a cost function satisfying a monotonicity condition.

1. Introduction

Dynamic programming has proven to be one of the principal methods for the formulation and solution of discrete optimization problems. A number of studies have explored the extent to which dynamic programming is applicable to such problems, including (Mitten 1964, Held and Karp 1967, Elmaghraby 1970, Bonzon 1970, Ibaraki 1972,1973, and other cited in the references). A recent survey of solution techniques and applications of dynamic programming appears in (Morin 1978). Mitten was the first to point out the essential role that the monotonicity of the cost function plays in a dynamic program. Subsequently, (Held and Karp 1967) studied dynamic programs in terms of a finite state machine with a superimposed cost structure (an sdp as defined below), and attacked the problem of characterizing the representations of a discrete optimization problem by a sdp with a monotonic cost function.

In this paper the notion of a discrete optimization problem is formalized as a discrete decision problem (ddp) and the general setting within which the functional equations of dynamic programming can be applied is formalized as a sequential decision process (sdp) following along the general lines of (Karp and Held 1967). Necessary and sufficient conditions for the representation in two different senses of a ddp by a sdp are established in theorems 2 through 7. In the first sense (a strong representation) the set of all optimal solutions to the discrete optimization problem is obtainable from the solution of the functional equations of dynamic programming. In the second sense (a weak representation) a nonempty subset of optimal solutions is obtainable from the solution of the functional equations of dynamic programming. It is shown that the well known principle of optimality corresponds to a strong representation. A more general version of the principle of optimality is given which corresponds to a weak representation of a ddp by a sdp. It is shown that sdp's having a strictly monotonic cost function are in one to one correspondence with strong representations of ddp's. Finally a new derivation is given of the result that sdp's having a monotonic cost function are in one-to-one correspondence with weak representations of a ddp.

Our notion of a weak representation is new in that we neither require all optimal solutions nor the correct cost of the optimal solutions, but are satisfied with some optimal solutions. Presumeably if the correct costs were required, one could compute the cost of an optimal solution using the cost function of the ddp after they have been found by some method. The notion of strong representation was introduced, along with an even stronger sense of representation, in (Ibaraki 1972).

2. Definitions.

A discrete decision problem is intended as a general model of combinatorial optimization problems. A discrete decision problem is a system $D=(A,S,P,f)$ where

A is a finite nonempty alphabet (set of primitive decisions),

$S \subseteq A^*$ (set of feasible policies),

P is a set (the set of data inputs for the problem),

$f: S \times P \to R$ where R is the set of positive reals, (cost or objective function).

An instance of a discrete decision problem D, denoted D(p), is given by a particular data input $p \in P$. A policy $s \in S$ is optimal with respect to input $p \in P$ if $\forall t \in S$ $f(s,p) \leq f(t,p)$. The set of optimal policies for the problem instance D(p) is denoted O(D,p). We will be interested in the conditions under which the problem of finding O(D,p) or a subset of O(D,p) can be formulated by a dynamic program.

One of the simplest discrete decision problems is the problem of finding the least cost path from the start node to a goal node in an arc-weighted directed graph. This problem can be represented as a ddp as follows; let A be the set of arcs (i,j) in the graph where (i,j) represents the decision to move from node i to node j, S is then the set of sequences of arcs which move from the start node to a final node, P is the set of cost matrices $(p_{i,j})$ where $p_{i,j}$ is the cost of arc (i,j), and finally f(s,p) is the cost of arc sequence (path) s with respect to input p; more precisely, $f(s,p) = \sum_{(i,j) \in s} p_{i,j}$.

The functional equations of dynamic programming apply to a kind of process called a sequential decision process. A <u>sequential decision process</u> (<u>sdp</u>) is a system $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ where

A is a finite nonempty alphabet (set of primitive decisions),

Q is a set (set of states),

$q_0 \in Q$ (start state),

$Q_f \subseteq Q$ (set of final states),

$t:QxA \rightarrow Q$ (transition function),

$h:RxQxAxP \rightarrow R$ (cost or objective function),

$k:P \rightarrow R$ (initial cost function),

P is a set (input data specifications).

The transition function t applies a decision $a \in A$ to a state $q \in Q$ resulting in a transition to a new state $t(q,a)$. We can extend the domain of t to QxA^* by the following recursive definition: let $t(q,e)=q$ for $q \in Q$, where e is the empty sequence, $t(q,xa)=t(t(q,x),a)$ for $q \in Q$, $x \in A^*$, and $a \in A$. Thus $t(q,xa)$ is the state resulting from applying the decision sequence xa to the initial state q. When only one argument is given to t the path will be assumed to originate at the start state, thus $t(x)$ is the state resulting from applying the decision sequence x from the start state. Let $F(\Pi)=\{x|t(x) \in Q_f\}$. $x \in F(\Pi)$ is a <u>feasible decision sequence</u> which t maps (by definition) from q_0 to some final state $q_f \in Q_f$. Note that the first five components of a discrete decision problem comprise a finite state automaton (Hopcroft and Ullman 1969). The cost function $h(c,q,a,p)$ is the cost of reaching state $t(q,a)$ by a sequence reaching state q with cost c which is extended by decision a. The initial cost function $k(p)$ is the cost of a null sequence given input p. It will be useful to consider the special case of decision sequences applied to the start state as follows: let $g(e,p)=k(p)$, $g(xa,p) = h(g(x,p),t(x),a,p)$ for $x \in A^*$, $a \in A$, $p \in P$. Thus $g(x,p)$ gives the cost of reaching state $t(x)$ from q_0 by means of the sequence of decisions x. Finally since we are interested in optimal decision sequences let us define (and assume the existence of) $G(q_s,p)=k(p)$ and $G(q,p) = \min_{\{x \mid t(x)=q\}} g(x,p)$ for all $q \neq q_s$, $p \in P$, thus $G(q,p)$ is the cost of the least cost decision sequence reaching state q from q_0. We say $x \in A^*$ is an <u>optimal decision sequence reaching state q</u> if $t(x)=q$ and $G(q,p)=g(x,p)$. The set of optimal decision sequences reaching a final state of Π are denoted $O(\Pi,p)$. Note that $O(\Pi,p)$ is always nonempty since there is at least one least cost sequence reaching each final state of Π. A sdp Π <u>represents</u> a ddp D if $F(\Pi)=S$ and $O(\Pi,p) \subseteq O(D,p)$.

3. <u>Representations of a discrete decision problem.</u>

Before turning to our primary problem of characterizing the representations of a ddp by a dynamic program, we give necessary and sufficient conditions for the representation, as defined above, of a ddp by an sdp. We first summarize some concepts and results on finite automata (Hopcroft and Ullman 1969) which will be needed only in the present section. The <u>equiresponse</u> <u>relation</u> of a finite automaton is defined by the relation xRy iff $t(x)=t(y)$ for all $x,y \in A^*$. An equivalence relation R on A^* is called <u>right</u> <u>invariant</u> if $xRy \rightarrow (\forall z \in A^*) xzRyz$. If R and T are equivalence relations on A^* then R <u>refines</u> T if $\forall x,y \in A^*$ $xRy \rightarrow xTy$. An equivalence relation has <u>finite</u> <u>rank</u> if it has only a finite number of equivalence classes. Note that the equiresponse relation on a finite automaton is right invariant since $t(x)=t(y) \rightarrow t(xz) = t(t(x),z) = t(t(y),z) = t(yz)$. Finally for some $S \subseteq A^*$ define the equivalence relation R_S as follows:

$$xR_Sy \text{ iff } (\forall z \in A^*)\ xz \in S \leftrightarrow yz \in S.$$

The following lemma gives us an essential property of finite automata.

<u>Proposition</u> <u>1</u>. Let $S \subseteq A^*$ and let R be a right invariant equivalence relation of finite rank, then R is the equiresponse relation of a finite automaton which accepts S iff R refines R_S.

proof: see (Hopcroft and Ullman 1969; pp 29).

<u>Theorem</u> <u>1</u>. A sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ represents a ddp $D=(A,S,P,f)$ iff the following conditions hold:

1. the equivalence relation R defined by xRy iff $t(x)=t(y)$ for $x,y \in A^*$ is a right invariant equivalence relation of finite rank which refines R_S.
2. $(\forall p \in P)(\exists x \text{ s.t. } t(x) \in Q_f)(\forall y \text{ s.t. } t(y) \in Q_f)\ g(y,p) \leq g(x,p) \rightarrow y \in O(D,p)$.

proof: (if): Suppose that conditions 1 and 2 hold. By proposition 1, R is the equiresponse relation of a finite automaton which accepts the language S, so $F(\Pi)=S$. Let $\hat{x}$ satisfy condition 2, so $(\forall y \in S$ s.t. $t(y)\in Q_f)$ $g(y,p)\leq g(\hat{x},p) \rightarrow y\in O(D,p)$. Let $\hat{y}\in O(\Pi,p)$ so $(\forall y$ s.t. $t(y)\in Q_f)$ $g(\hat{y},p)\leq g(y,p) \rightarrow g(\hat{y},p)\leq g(\hat{x},p) \rightarrow \hat{y}\in O(D,p)$ thus $O(\Pi,p)\subseteq O(D,p)$.

(only if): Suppose now that Π represents D, so $F(\Pi)=S$ and $O(\Pi,p)\subseteq O(D,p)$. R is the equiresponse relation of a finite automaton which accepts S, so R is a right invariant equivalence relation of finite rank. By proposition 1, R refines R_S, so condition 1 holds. Let $\hat{y}\in O(\Pi,p)$ then $(\forall y$ s.t. $t(y)\in Q_f)$ $g(y,p)\leq g(\hat{y},p) \rightarrow g(y,p) = g(\hat{y},p) \rightarrow y\in O(\Pi,p) \rightarrow y\in O(D,p)$. Thus condition 2 holds. QED

There are several important aspects to our representations of ddp's by sdp's which should be pointed out. In mapping from a ddp to a sdp, we assume the notion of a state (the equivalence classes of R in theorem 1), the existence of the transition function t which only depends on the current state and input decision, and a cost function which is separable in the sense that the cost of adding a transition onto the end of a sequence only depends on the current state, the input decision, and the cost of the sequence (in general the cost might depend on all previous decisions). This much structure is implicit in the concept of a dynamic program. A closer examination of these assumptions may be found in (Elmaghraby 1970).

4. Strong representations of a discrete decision problem.

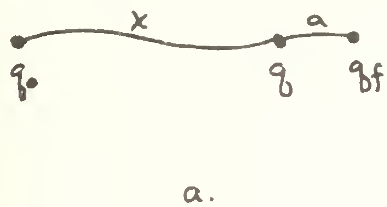
Our purpose is to discover the conditions under which a sdp Π represents a ddp D by means of a discrete dynamic program. The principal underlying dynamic programming has been formulated by Bellman in the Principle of Optimality (Bellman 1957) and can be paraphrased as follows:

An optimal sequence has the property that no matter what the next-to-last state and the next-to-last decision are the sequence reaching the next-to-last state must be optimal.

This version of the principle of optimality is illustrated in figure 1a. If for $a \in A$, $x \in A^*$ xa is an optimal sequence from state q_0 to q_f then x is an optimal sequence from q_0 to q. In general the principle of optimality implies that if xy, for $x,y \in A^*$, is an optimal sequence from q_0 to q_f then x is an optimal sequence from q_0 to $t(q_0,x)$ and y is an optimal sequence from $t(q_0,x)$ to q_f as illustrated in figure 1b. This illustration applies only to discrete sequences and so should not be construed to demonstrate the full range of dynamic programming which is much broader.

In terms of an sdp the principle of optimality can be made precise as follows:

$$(\forall p \in P)(\forall x \in A^*)(\forall a \in A)\ G(t(xa),p)=g(xa,p) \rightarrow G(t(x),p)=g(x,p) \quad (1)$$



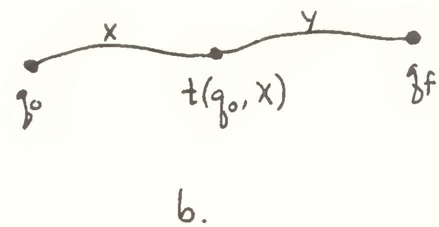


Figure 1.

The following lemma states an equivalent form for (1). Let $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ be a sdp. h is s'-monotonic if for all states $q \in Q$, optimal sequences xa reaching state q, and sequences ya reaching q, we have $g(x,p)<g(y,p) \leftrightarrow g(xa,p)<g(ya,p)$. A sdp containing a s'-monotonic cost function is a s'-monotonic sequential decision process (s'-msdp). We say h is strictly monotonic (s-monotonic) if for all $x,y \in A^*$ such that $t(x)=t(y)$, $g(x,p)<g(y,p) \rightarrow g(xa,p)<g(ya,p)$. A sequential decision process which contains a s-monotonic cost function is called a strictly monotonic sequential decision process (s-msdp).

Theorem 2. (1) holds for an sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff h is s'-monotonic.

proof: (only if): Suppose that (1) holds for some sdp Π and that h is not s'-monotonic. Let xa be an optimal sequence reaching state q and let y be a sequence such that $t(x)=t(y)$. Suppose first that $g(x,p)<g(y,p)$ and $g(xa,p)\geq g(ya,p)$. Since $G(q,p)=g(xa,p)\geq g(ya,p)$, we have $g(xa,p)=g(ya,p)$. By (1),

$G(q',p)=g(x,p)=g(y,p)$, but this contradicts our assumption that $g(x,p)<g(y,p)$. Thus $g(x,p)<g(y,p) \rightarrow g(xa,p)<g(ya,p)$. Suppose instead we have $g(xa,p)<g(ya,p)$ but $g(x,p)\geq g(y,p)$. $g(x,p)\neq g(y,p)$ since $g(xa,p)\neq g(ya,p)$ so $g(x,p)>g(y,p)$. But by (1) and our assumption that xa is an optimal sequence reaching q, we have $G(q',p)=g(x,p)\leq g(y,p)$ by definition of G. This contradiction shows that $g(xa,p)<g(ya,p) \rightarrow g(x,p)<g(y,p)$ when x is an optimal sequence reaching state q. Thus (1) $\rightarrow$ h is s'-monotonic.

(if): Suppose now that h is s'-monotonic. If (1) does not hold then for some sequence xa such that $t(xa)=q$, we have $G(q,p)=g(xa,p)$ but $G(q',p)\neq g(x,p)$ where $t(q',a)=q$. For some $y\in A^*$ such that $t(x)=t(y)$ we have $G(q',p)=g(y,p)<g(x,p)$. If $g(ya,p)=g(xa,p)=G(q,p)$ then h is not s'-monotonic (with respect to optimal sequence xa), so we must have $g(ya,p)>g(xa,p)$. But since h is s'-monotonic we have $g(y,p)>g(x,p)$ which contradicts our earlier finding that $g(y,p)<g(x,p)$. Thus (1) must hold. QED

In practice we wish to find optimal policies between states. We define below the tables $T(q,p)$ which store the information necessary to obtain optimal policies. Formally for all $q\in Q, p\in P$ $T(q,p)$ is a subset of $Q\times A$. ($T: Q\times P \rightarrow 2^{Q\times A}$). A set of policies $\theta(q,p)$ are <u>obtainable</u> from the tables $T(q,p)$ as follows: let

$$\theta(q_s,p) = \{(q_s,e)\}, \quad \text{where e is the empty string,}$$

$$\theta(q,p) = \{ya \mid (q',a)\in T(q,p) \text{ and } y\in\theta(q',p)\} \quad \text{for } q\neq q_s.$$

A ddp $D=(A,S,P,f)$ is <u>strongly-represented</u> (<u>weakly-represented</u>) by a sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ if i) Π represents D, ii) the functional equations (2) and (3) given below hold and iii) for $q\in Q$, $p\in P$ the set of policies obtainable from the tables $T(q,p)$ is the set (subset) of all optimal policies; in particular $\bigcup_{q\in Q_f} \theta(q,p)=O(\Pi,p)$ ($\bigcup_{q\in Q_f} \theta(q,p)\subseteq O(\Pi,p)$ for a weak representation).

$$G(q_s,p)=k \tag{2}$$

$$G(q,p)=\min_{\{(q',a)\mid t(q',a)=q\}} h(G(q',p),q',a,p) \tag{3}$$

$$T(q,p)=\{(q',a)\mid t(q',a)=q,\ G(q,p)=h(G(q',p),q',a,p)\} \tag{4}$$

Note that if Π strongly (weakly) represents D then by (i) $O(\Pi,p)=O(D,p)$ and thus $\bigcup_{q\in Q_f} \theta(q,p) = O(D,p)$ ($\bigcup_{q\in Q_f} \theta(q,p)\subseteq O(D,p)$) i.e., the construction of the tables θ by means of (2),(3), and (4) results in the construction of all (a nonempty subset of) optimal solutions to the ddp D.

<u>Lemma 1</u>. $x\in\theta(q,p) \rightarrow x$ is an optimal sequence reaching state q.

proof: the lemma follows immediately from the stronger lemma 2 which is given in the appendix.

We do not require that an optimal sequence have the same cost in D as in Π. Our interest is in obtaining optimal solutions and in making use of the functional equations (2) and (3). These equations are characteristic of dynamic programming and are often considered a direct translation of the principle of optimality. We take (1) as a more direct translation and show next that in the sense of a strong representation (1) and the equations (2) and (3) are equivalent.

<u>Theorem 3</u>. A ddp $D=(A,S,P,f)$ is strongly-represented by an sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff Π represents D and (1) holds.

proof: (if): Suppose that (1) holds and Π represents D. In order to show that the ddp D may be strongly-represented by an sdp Π, we must show that Π represents D (which we have assumed), (2) and (3) hold, and that all optimal policies may be obtained from the tables defined by (4). First, (2) holds by definition of G. Let $H(q,p)$ denote the right hand side of (3). We will show that $G(q,p)=H(q,p)$. Suppose that ya is an optimal policy reaching state q, so $G(q,p)=g(ya,p)$. Since (1) holds we then have $G(\hat{q},p)=g(y,p)$ where $t(\hat{q},a)=q$. Thus $G(q,p) = h(g(y,p),\hat{q},a,p) = h(G(\hat{q},p),\hat{q},a,p) \geq \min_{\{(q',a)|t(q',a)=q\}} h(G(q',p),q',a,p) = H(q,p)$, or $G(q,p)\geq H(q,p)$.

Now let $H(q,p)=h(G(\hat{q},p),\hat{q},a,p)$ for some $\hat{q}\in Q$ and suppose $G(\hat{q},p)=g(y,p)$ where $t(y)=\hat{q}$. i.e., y is an optimal policy reaching $\hat{q}$. Let $t(ya)=q$ then $G(q,p)\leq g(ya,p) = h(g(y,p),\hat{q},a,p) = h(G(\hat{q},p),\hat{q},a,p) = H(q,p)$, thus $G(q,p)\leq H(q,p)$. Combining these results we have $G(q,p)=H(q,p)$ and (3) holds.

By lemma 1 all policies in $\Theta(q,p)$ are optimal with respect to h. Suppose though that not all optimal policies can be obtained from (4). Let xa be an optimal policy of shortest length reaching state q which is not in $\theta(t(xa),p)$. Let $t(x)=q'$. By (1) x is optimal thus $x \in \theta(q',p)$ (since x has shorter length than xa) and $G(q',p)=g(x,p)$. Since $xa \notin \theta(t(xa),p)$ we must have $G(t(xa),p) < h(G(q',p),q',a,p) = h(g(x,p),q',a,p) = g(xa,p)$, but this contradicts our assumption that xa is an optimal sequence reaching state q. Therefore $(q',a) \in T(q,p)$ and by definition $xa \in \theta(q,p)$, so $\theta(q,p)$ is the set of all optimal sequences reaching state q. In particular $\bigcup_{q \in Q_f} \theta(q,p) = O(\Pi,p)$.

(only if): Suppose now that the ddp D is strongly-representable by the sdp Π. For some $q \in Q$, $x \in A^*$ we are able to obtain all optimal policies reaching state q using (2), (3), and (4). consider $xa \in \theta(q,p)$ where $t(xa)=q$, $t(x)=q'$. By lemma 1 xa is an optimal sequence reaching state q. By definition $x \in \theta(q',p)$, and by lemma 1 x is an optimal policy reaching q', so $G(q',p)=g(x,p)$. Thus (1) holds. Π represents D by assumption. QED

<u>Corollary 1</u>. A ddp $D=(A,S,P,f)$ is strongly-represented by a sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff Π represents D and Π is a s'-msdp.

proof: immediate from theorems 2 and 3.

The s'-monotonicity of the cost function of an sdp is an essential ingredient in a strong representation of a ddp. It can be shown however that any s'-monotonic cost function is effectively equivalent to some strictly-monotonic cost function.

Given a s'-monotonic function h, define the function g' (and thereby h' implicitly) as follows:

$$g'(xa,p) = \begin{cases} g(xa,p) & \text{if } G(q,p)=g(xa,p) \\ G(q,p)+g'(x,p) & \text{otherwise.} \end{cases} \tag{5}$$

Define $G'(q,p) = \min_{\{q|t(x)=q\}} g'(x,p)$. Note that by definition $G(q,p)=G'(q,p)$ for all states q and inputs p. Lemma 4 given in the appendix establishes the effective equivalence of h and h' in the sense that the set of optimal sequences obtained for each state is the same for both cost functions.

<u>Lemma</u> <u>3</u>. If h is s'-monotonic then h' defined by (5) is strictly monotonic.

proof: Let h' be defined from the s'-monotonic function h by (5). Suppose for $x,y \in A^*$ such that $t(x)=t(y)$, we have $g'(x,p)<g'(y,p)$. We have 2 cases to consider in order to show that $g'(xa,p)<g'(ya,p)$. Let $a \in A$ such that $t(xa)=q$. Case 1: ya is not optimal. By construction of g', $g'(ya,p)=G(q,p)+g'(y,p)$ and $g'(xa,p)$ has the value $G(q,p)$ or $G(q,p)+g'(x,p)$ either of which is strictly less than $g'(ya,p)$. Case 2: ya is an optimal sequence reaching state q. If ya is optimal then $g'(ya,p)=g(ya,p)=G(q,p)$. Also by theorem 2, (1) holds so y is an optimal sequence; i.e., $g'(y,p) = g(y,p) = G(q',p) = G'(q',p)$, but this contradicts our assumption that $g'(x,p)<g'(y,p) = G'(q',p)$. QED

Theorem 4. A ddp $D=(A,S,P,f)$ is strongly represented by a sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff there is a strictly monotonic sdp $\Pi'=(A,Q,q_0,Q_f,T,h',k,P)$ which strongly represents D.

proof: (only if): Clearly any s-msdp is an s'-msdp so by corollary 1 the statement of the theorem is consistent and D is strongly represented by Π'.

(if): Suppose that D is strongly represented by $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$, then by corollary 1 h is a s'-monotonic cost function. Consider h' defined by (5) which is s-monotonic by lemma 3. We need to show that $\Pi'=(A,Q,q_0,Q_f,T,h',k,P)$ strongly represents D. (2) holds by definition. In order to show that (3) holds, let xa be an optimal sequence reaching state q. By construction $G(q,p)=G'(q,p)$ for all states $q \in Q$. Equation (3) then holds for G' since it holds for G by corollary 1. Equation (4) holds since lemma 4, given in the appendix, shows that $\theta'(q,p)=\theta(q,p)$ so $\theta'(q,p)$ is the set of all optimal sequences reaching state q. Finally Π' represents D since $F(\Pi')=F(\Pi)=S$ and $O(D,p)=O(\Pi,p) = \bigcup_{q \in Q_f} \theta(q,p) = \bigcup_{q \in Q_f} \theta'(q,p) = O(\Pi',p)$. QED

5. Weak representations of a discrete decision problem.

We have been looking at the conditions under which we can find all optimal decision sequences reaching any state from q_0. In practice we may relax this requirement and be satisfied with some (or just one) optimal sequences to each state in Q. We now explore the conditions under which this requirement can be satisfied.

We have seen how a direct translation of the principle of optimality helped to establish the conditions for its application. In the more general situation faced now it may be helpful to give a generalized principle of optimality which applies when we are interested in obtaining only some optimal decision sequences.

Generalized principle of optimality (forward version): If there is an optimal sequence reaching state q, then there is an optimal sequence reaching state q with the property that no matter what the last decision and last state q' were, the sequence reaching q' is an optimal sequence.

Given $p \in P$, a sequence xa is 1-optimal if $G(t(xa),p)=g(xa,p)$ and $G(t(x),p)=g(x,p)$. This generalized principle of optimality can be formalized as follows:

$(\forall p \in P)(\forall q \in Q)$ there is a 1-optimal sequence reaching state q (6)

In these terms we can reformulate the (original) principle of optimality as follows: $\forall p \in P$ $\forall q \in Q$ every optimal sequence reaching state q is 1-optimal. Condition (6) can be expressed soley in terms of the cost function h as given below in theorem 5. h is b-monotonic if for all $q \in Q$, some optimal sequence xa reaching q, and sequence $ya \in A^*$ reaching q, we have $g(xa,p)<g(ya,p) \rightarrow g(x,p)<g(y,p)$. A sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ in which h is b-monotonic is a b-monotonic sequential decision process (b-mdsp).

Theorem 5. (6) holds iff h is b-monotonic.

proof: (if): Consider an arbitrary state $q \in Q$ and let h be b-monotonic. We will show there exists a 1-optimal sequence reaching state q. Let xa be an optimal sequence reaching state q. Let $P(q)$ denote the set of sequences such that $y \in P(q)$ iff $t(y)=q$. Partition $P(t(x))$ into two sets as follows: let

$$Y(x,a) = \{y \mid y \in P(t(x)),\ g(xa,p)=g(ya,p),\ g(x,p)>g(y,p)\}$$

$$Z(x,a) = \{z \mid z \in P(t(x)),\ g(xa,p)<g(za,p)\} \cup$$

$$\{z \mid z \in P(t(x)),\ g(xa,p)=g(za,p),\ g(x,p)\leq g(z,p)\}$$

For any $z \in Z(x,a)$ we have $g(x,p)\leq g(z,p)$, either by the monotonicity of h in the case that $g(xa,p)<g(za,p)$ or by definition in the other case. Thus if $Y(x,a)$ is empty then $G(t(x),p)=g(x,p)$ and xa is a 1-optimal sequence reaching state q. On the other hand if $Y(x,a)$ is nonempty, we have $y' = \min_{y \in Y(x,a)} g(y,p)$ for some $y' \in Y(x,a)$. Then $g(y',p)\leq g(y,p)$ for all $y \in Y(x,a)$, and $g(y',p)<g(x,p)\leq g(z,p)$ for all $z \in Z(x,a)$, thus $G(t(x),p)=g(y',p)$. But $g(y'a,p)=g(xa,p)=G(q,p)$, so $y'a$ is a 1-optimal sequence reaching state q.

(only if): Suppose now that (6) holds. For an arbitrary state q, let $G(q,p)=g(xa,p)$ and $G(q',p)=g(x,p)$ where $t(q',a)=q$ and $t(x)=q'$; i.e., xa is 1-optimal sequence reaching state q. Suppose that h is not b-monotonic, so for some sequence ya we have $g(xa,p)<g(ya,p)$ and $g(x,p)\geq g(y,p)$. By the 1-optimality of xa we have $g(x,p)=G(q',p)\leq g(y,p)$. Furthermore we must have $g(x,p)<g(y,p)$ since $g(x,p)=g(y,p) \rightarrow h(g(x,p),t(x),a,p) = h(g(y,p),t(x),a,p)$; i.e., $g(xa,p)=g(ya,p)$. This contradiction

shows that h is b-monotonic. QED

Theorem 6. A ddp $D=(A,S,P,f)$ is weakly-represented by a sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff Π represents D and (6) holds.

proof: (if): Suppose that the ddp $D=(A,S,P,f)$ is weakly-represented by a sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$. By definition Π represents D. Now let q be an arbitrary state. By (2), $G(q,p)=\min_{\{(q',a)\,|\,t(q',a)=q\}} h(G(q',p),q',a,p)$. Let $G(q,p) = h(G(\hat{q},p),\hat{q},a,p)$ and let $G(\hat{q},p)=g(y,p)$, then $G(q,p) = h(G(\hat{q},p),\hat{q},a,p) = h(g(y,p),\hat{q},a,p) = g(ya,p)$. We have just shown that ya is a 1-optimal sequence reaching state q. Thus (6) holds.

(only if): Suppose now that Π represents D and (6) holds. For any state $q \in Q$, there exists a sequence xa such that $t(xa)=q$, $G(q,p)=g(xa,p)$, and $G(\hat{q},p)=g(x,p)$. $G(q,p) = g(xa,p) = h(g(x,p),\hat{q},a,p) = h(G(\hat{q},p),\hat{q},a,p)$ which implies that we can find the value $G(q,p)$ by minimizing the expression $h(G(q',p),q',a,p)$ over all $q' \in Q$, $a \in A$ such that $t(q',a)=q$, and thus we get (3). (2) follows by definition. By definition all elements of $\Theta(q,p)$ are optimal sequences which reach state q. To see that $\Theta(q,p)$ is nonempty, note that since (6) holds there is a sequence xa such that $G(q,p)=g(xa,p)$ and $G(q',p)=g(x,p)$ where $T(q',p)=q$ and by definition such an xa is in $\Theta(q,p)$. Finally Π represents D by assumption. QED

Corollary 2. A ddp $D=(A,S,P,f)$ is weakly-representable by a sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff Π represents D and Π is a b-msdp.

proof: immediate from theorems 5 and 6.

We have now characterized the classes of sdp's which weakly and strongly represent ddp's. The difference between these two types of representations is illustrated in figure 2. Here h is b-monotonic but h is not s'-monotonic. According to equation (3), in order to determine an optimal sequence reaching q, we consider an extension of an optimal sequence reaching q'. But in restricting the search to optimal sequences reaching q', equation (3) overlooks the optimal sequence ya reaching q. This illustrates why b-msdp's can only weakly-represent a ddp.

The conditions established for the weak-representation of a ddp are necessary in order to take care of fairly pathological cost functions. It can be shown however that the cost function of any sdp which weakly represents is equivalent to other cost

---

$g(x,p)=10$ $g(xa,p)=16$
$g(y,p)=12$ $g(ya,p)=16$

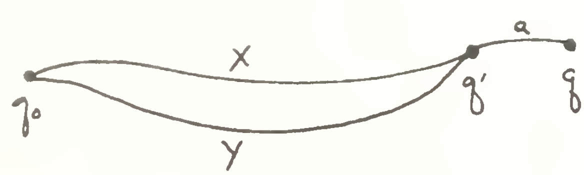


Figure 2.

---

functions with nicer properties. Given a cost function h which is b-monotonic, define the function g' (and thereby h') as follows:

$$g'(x,p)=\begin{cases} g(x,p) & \text{if } x \text{ is 1-optimal} \\ G(t(x),p)+1 & \text{otherwise.} \end{cases} \qquad (7)$$

Define $G'(q,p)=\min\limits_{t(x)=q} g'(x,p)$. Lemma 4 given in the appendix establishes the effective equivalence of h and h' in the sense that the set of optimal sequences obtained for each state is the same for both cost functions.

h is <u>monotonic</u> if $\forall x,y \in A^* \; \forall a \in A$ such that $t(x)=t(y)$ $g(x,p) \leq g(y,p) \rightarrow g(xa,p) \leq g(ya,p)$. An sdp with cost function h which is monotonic is a <u>monotonic</u> <u>sequential</u> <u>decision</u> <u>process</u> (<u>m</u>-<u>sdp</u>).

<u>Lemma</u> <u>5</u>. If for some sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ h is b-monotonic then h' defined by (7) is monotonic.

proof: Consider the function h' defined in (7). h' can be shown to be monotonic as follows. Let $t(x)=t(y)=q'$, $t(q',a)=q$ and $g'(x,p) \leq g'(y,p)$. If xa is 1-optimal then $g'(xa,p)=g(xa,p)=G(q,p)$ and since $g'(ya,p)$ has the value $G(q,p)$ or $G(q,p)+1$, $g'(xa,p) \leq g'(ya,p)$. Suppose now that ya is 1-optimal, then $G(q,p)=g(ya,p)$ and $G(q',p)=g(y,p)$, $g'(ya,p)=g(ya,p)$ and $g'(y,p)=g(y,p)=G(q',p)=g'(x,p)$ (since $g'(x,p) \leq g'(y,p)$. But if $g'(x,p)=g'(y,p)$ then $g'(xa,p) = h'(g'(x,p),q',a,p) = h'(g'(ya,p),q',a,p) = g'(y,p)$ (thus $g'(xa,p) \leq g'(ya,p)$). If neither xa nor ya is 1-optimal then $g'(xa,p)=g'(ya,p)=G(q,p)+1$. In all cases the monotonicity of h' is shown. QED

The following result is well known (Elmaghraby 1970, Bonzon 1970) in the sense that dynamic programs are in one to one correspondence with monotonic sdp's. However to the author's knowledge it has not been pointed out that m-sdp's can only weakly represent a ddp; i.e., one is not guaranteed to be able to obtain all optimal solutions from a representation by a m-sdp.

<u>Theorem 7</u>. A ddp $D=(A,S,P,f)$ is weakly-represented by some sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$ iff there is a m-sdp $\Pi'=(A,Q,q_0,Q_f,T,h',k,P)$ which weakly-represents D.

proof: (if): We must show that a m-sdp can represent D. Let xa be an optimal sequence reaching q, so $G(q,p)=g(xa,p)$. Suppose $g(xa,p)<g(ya,p)$ yet $g(x,p)\geq g(y,p)$. By the monotonicity of h', we get $g(xa,p)\geq g(ya,p)$ which contradicts our assumption. Thus $g(x,p)>g(y,p)$ and h' is b-monotonic. By corollary 2, Π' weakly-represents D.

(only if): Suppose that D is weakly-represented by an sdp $\Pi=(A,Q,q_0,Q_f,T,h,k,P)$, and h' is defined by (7) from h, then by corollary 2, h is b-monotonic and by lemma 5 h' is monotonic.

We can show that D is weakly-represented by the sdp $\Pi'=(A,Q,q_0,Q_f,T,h',k,P)$. (2) holds by definition. Let $x\in A^*$ be a 1-optimal sequence reaching state $q\in Q$ so $G(q,p)=g(x,p)$. Such a sequence exists by theorem 6. By construction $g'(x,p)=g(x,p)$ so

$G'(q,p)=G(q,p)$ for all states $q \in Q$. Equation (3) must hold for $G'(q,p)$ since it holds for $G(q,p)$ as a result of corollary 2. Lemma 4 shows that $\theta(q,p)=\theta'(q,p)$ so $\theta'(q,p)$ is a nonempty subset of optimal sequences. Finally Π' represents D since $F(\Pi')=F(\Pi)=S$ and $O(\Pi',p) = \bigcup_{q \in Q_f} \theta'(q,p) = \bigcup_{q \in Q_f} \theta(q,p) = O(\Pi,p) \subseteq O(D,p)$. QED

6. Conclusion.

This paper has given necessary and sufficient conditions for the strong and weak representation of a discrete decision problem by a sequential decision process. Strictly monotonic (monotonic) sequential decision processs have been shown to be equivalent in the strong (weak) representation sense to the class of discrete decision problems which can be formulated as discrete dynamic programs. We have shown that the problems to which the principle of optimality applies are a subclass of the problems to which the functional equations of dynamic programming are applicable.

Appendix

In order to establish lemma 1 we will need the following definition and lemma. We say $x \in A^*$ is completely-optimal if every initial segment (every $y \in A^*$ such that there exists $z \in A^*$ such that $yz=x$) y of x is 1-optimal.

Lemma 2. $xa \in \theta(q,p)$ iff xa is completely optimal.

proof: by induction on the length of a sequence. Let the length of x be 1, i.e., $x \in A$. $(q_s, x) \in \theta(q,p)$ iff $x \in T(q,p)$ and $e \in \theta(q,p)$ where e is the empty sequence and $t(x)=q$. By definition $e \in \theta(q_s,p)$ and $x \in \theta(q,p)$ iff $G(q,p)=g(x,p)$ iff x is an optimal sequence.

Induction step: Assume that the lemma holds for any sequence of length $<m$ and let the length of the sequence xa be m. $xa \in \theta(q,p)$ iff $(q',p) \in T(q,p)$ and $x \in \theta(q',p)$ where $T(q',p)=q$. By induction hypothesis $x \in \theta(q',p)$ iff x is completely optimal. This implies that $G(q',p)=g(x,p)$. Also $(q',p) \in T(q,p)$ iff $G(q,p)=h(G(q',p),q',a,p) = h(g(x,p),q',a,p)=g(xa,p)$. ($xa$ is 1-optimal and x is completely optimal $\rightarrow xa$ is completely optimal). i.e., xa is completely optimal. QED

The following lemma establishes the effective equivalence of h and h' defined by (5) in the sense that the set of optimal sequences obtained for each state is the same for both cost functions. The lemma also holds true for h' defined by equation (7).

Lemma 4. $\forall q \in Q, \forall p \in P\ \theta(q,p)=\theta'(q,p)$.

proof: $x \in \theta(q,p)$ iff x is completely optimal (by lemma 2),

iff $x=a_1a_2\ldots a_n$ and $a_1\cdots a_i$ is 1-optimal with respect to h for $i=1,\ldots,n$

iff $g'(a_1,p)=g(a_1,p)=G'(t(a_1),p)$ and ... and $g'(a_1\ldots a_n,p) = g(a_1\ldots a_n,p) = G(t(a_1\cdots a_n),p)$ by construction,

iff x is completely optimal with respect to h',

iff $x \in \theta'(q,p)$ (by lemma 2).

QED

References

Bellman, R.E. (1957), Dynamic Programming. Princeton University Press, Princeton, NJ, 1957.

Bonzon, P. (1970), Necessary and Sufficient Conditions for Dynamic Programming of Combinatorial Type. J. of ACM 17(4), October, 1970, 675-682.

Denardo, E.V. (1967), Contraction Mappings in the Theory underlying Dynamic Programming. SIAM Review 9(2), April, 1967, 165-177.

Denardo, E.V., and Fox, B.L. (1979), Shortest-Route methods: 1. Reaching, Pruning, and Buckets. Operations Research 27(1), 1979, 161-186.

Elmaghraby, S.E. (1970), The Concept of "State" in Discrete Dynamic Programming. Journal of Math. Anal. Appl. 29, 523-557, 1970.

Hopcroft, J.E., and Ullman, J.D. (1969), Formal Languages and Their Relation to Automata. Addison-Wesley Publishing Co., Reading, Mass., 1969.

Ibaraki, T. (1972), Representation Theorems for Equivalent Optimization Problems. Info. and Control 21, 1972, 397-435.

Ibaraki, T. (1973), Solveable Classes of Discrete Dynamic Programming. J. Math. Anal. Appl. (1973)

Karp, R.M., and Held, M. (1967), Finite-State Processes and Dynamic Programming. SIAM J. Appl. Math., Vol 15, No. 3, May 1967, pp693-718.

Larson, R.E., and Casti, J.L. (1978), Principles of Dynamic Programming, Part 1 Basic Analytic and Computational Methods. Marcel Dekker, Inc., New York, NY, 1978.

Mitten, L.G. (1964), Composition Principles for Synthesis of Optimal Multistage Processes. Operations Research 12, 1964, 610-619.

Morin T.L., and Marsten, R.E. (1976), Branch and Bound Strategies for Dynamic Programming. Operations Research 24, pg 611-629, 1976.

Morin, T.L. (1978), Computational Advances in Dynamic Programming. in Dynamic Progamming and Its Applications. Ed. M.L. Puterman, Academic Press, New York, NY, 1978,pp 53-90.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
Dudley Knox Library Code 0142 Naval Postgraduate School Monterey, CA 93940	2
Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93940	1
Chairman, Code 52Bz Department of Computer Science Naval Postgraduate School Monterey, CA 93940	40
Professor Douglas R. Smith, Code 52Sc Department of Computer Science Naval Postgraduate School Monterey, CA 93940	12